

JJ-11 468US

Claims

What is claimed is:

1. A method that allows SQL-like query selections with user-defined ordinary and group plug-ins functions operating on structured and semi-structured data files; the method comprising:
 - a) providing three interconnected independent components: a Query Filter, a Data Extractor Application, and a Database; the Query Filter being an SQL-like dialect for issuing and parsing relational queries, the Data Extractor Application being a computer program that extracts the data or obtains information about the supported Database schema, and returns the database schema or the given columns of data as one row of text and the Database being a local or distributed collection of structured or semi-structured data files; and
 - b) passing requests between the Query Filter and the Data Extractor Application and between the Data Extractor Application and the Database to allow the Query Filter to view the data of the Database matching the SQL-like query selection.
2. A method as claimed in claim 1 wherein the Query Filter passes a request to the Data Extractor Application, to list the Database schema information.
3. A method as claimed in claim 2 wherein the Database schema information is returned from the Data Extractor Application to the Query Filter.

JJ-11 468US

4. A method as claimed in claim 1 wherein the Query Filter passes a request to the Data Extractor Application, to obtain the values of the desired columns from the Database.

5. A method as claimed in claim 4 wherein the Data Extractor Application returns the request to the Query Filter, with the values of the desired columns selected from the Database.

6. A method as claimed in claim 5 wherein the Query Filter and filters the undesirable column values outputting the intended column values only.

7. A method as claimed in claim 1, wherein the Data Extractor Application supports two types of interface protocols: a first protocol to return information about the Database schema, and a second protocol to return contents of the desired Database columns.

8. A method as claimed in claim 1 wherein the Query Filter possesses an SQL-like grammar defined via Backus Naur Form.

9. A method as claimed in claim 1, wherein the Query Filter learns via a UNIX environment variable about the location of the Data Extractor Application.

10. A method as claimed in claim 1, wherein for the Data Extractor Application to be implemented in any practical programming language of choice.

JJ-11 468US

11. A method as claimed in claim 1, wherein the Query Filter grammar is expanded with user-defined ordinary and group plug-ins functions.

12. A method as claimed in claim 11 wherein the ordinary and group user-defined plug-ins functions are defined by N arguments in the module plugin.c.

13. A method as claimed in claim 12 wherein last call support with all null arguments to the user-defined group plug-ins functions in the module plugin.c, to warrant computation of the final result is provided.

14. A method as claimed in claim 12 wherein data types of all the arguments passed to the plug-ins or values returned from the plug-ins subroutines are pointers to characters.

15. A method as claimed in claim 11 wherein the plugin.c module is computed with user-defined ordinary and group plug-ins functions, and linked it with the Query Filter using the make utility.

16. A method as claimed in claim 1 wherein the Database comprises one or more of local or dispersed throughout a cyberspace structured or semi-structured data files.